AFRL-RI-RS-TR-2009-23
In-House Final Technical Report
January 2009

OZTURK ALGORITHM AUGMENTATIONS

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2009-23 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
STEVEN JOHNS, Chief Multi-Sensor Exploitation Branch	JOSEPH CAMERA, Chief Information & Intelligence Exploitation Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
JAN 09

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
May 07 – Dec 08

4. TITLE AND SUBTITLE

OZTURK ALGORITHM AUGMENTATIONS

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Steven Salerno

5d. PROJECT NUMBER
459E

5e. TASK NUMBER
PR

5f. WORK UNIT NUMBER
OJ

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/RIEC
525 Brooks Rd.
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIEC
525 Brooks Rd.
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2009-23

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED*. PA#88ABW-2009-0176*

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this effort, augmentations to the Ozturk Algorithm were investigated, including additional graphical representations and determination of data correlation effects as introduced by data filtering. A graphical user interface was developed to demonstrate the algorithm concepts.

15. SUBJECT TERMS

Ozturk Algorithm, Rank-ordered statistics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Andrew J. Noga
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (*Include area code)***
U	U	U	UU	32	N/A

Table of Contents

List of Figures

1.0 Introduction

This report covers work initiated in the Summer of 2006 and completed in the Summer of 2008 as a summer engineering aide. Work in the applications of ranked-ordered statistics as a method of signal data analysis was conducted. More specifically work was conducted on the Ozturk algorithm, which is considered to have demonstrated superior performance over histogram and simple parameter-based techniques [1-4]. The work includes the research and implementation of new techniques based on the Ozturk algorithm. A GUI was also created in order to house the previously done work, [5] and the newly created implementation tools.

1.1 Overview of the Ozturk Algorithm

The Ozturk algorithm is an identification algorithm that allows for a variety of tests that not only identify a best fit probability density function (PDF), but also analyzes the sample data. The core algorithm was developed by Aydin Ozturk [3], and further worked upon in concurrence with E.J. Dudewicz and others [4]. Prior work was carried out with the help of Syracuse University and the only documented implementation of the algorithm for real world use was associated with research conducted at the AFRL Sensors Directorate at the Rome Research Site.

The basis of the algorithm deals with constructing a null-linked graph of vectors from the rank ordered statistics of the sample data. The algorithm is scale and location invariant, which makes it particularly attractive. From this graph one can easily identify a distribution based on its endpoint, Q_n, and surrounding null points. From the ordered statistics other tests can be carried out, including but not limiting to, testing for Normality, parameter estimation, identifying the best fit distribution and the newest test, the detection of correlation in Gaussian data. The power of the test comes from the ability to use small sample sizes while still being able to correctly identify the best fit distribution. The entirety of the algorithm will be discussed in Section 2.

1.2 Summer Overview

Much of the work done the Summer of 2006 was based on the previous work done in the Summer of 2005. Since there was now a working GUI implementation of the Ozturk algorithm, it was time to both test the algorithm's capabilities and limits. In [1-4] there has been no mention of the result of correlated data. This produces a potential problem with real world data, since it is not always uncorrelated. When you have this real world data, how do you determine whether or not it is correlated, and whether or not the correlation is strong enough to affect the algorithm? There needs to be either a pre-processing step or test to determine if the data is correlated, and if it is how strongly it is correlated. The second major constraint is that, there are distributions that can mimic other distributions if the parameters are set in a specific manner. In this case the algorithm can only identify one distribution, while there may actually be multiple distributions the data could be constructed from. Both of these issues were the basis for most of the work, along with the full construction of a new Matlab GUI to provide the user the ability to implement the new tools.

2.0 PDF Identification & Analysis – Ozturk Algorithm

Most statistical tests that are used presently test a data set for a specific distribution. Often the test is designed specifically to assess the normality (Gaussian PDF is assumed) of a data set. However when the solution does not match the hypothesis no alternative possibility is given. This results in the tedious task of having to do multiple statistical tests in order to identify the best fit distribution. Also a majority of the tests start to lose their power when the sample size decreases. Both these common issues are addressed through the Ozturk algorithm [1-4]. The algorithm uses its own test statistics to create "linked vectors" to create a goodness-of-fit test for univariate and multivariate distributions.

2.1 Procedure – Ozturk Algorithm

General reference is made to [2, 3]. Let $X_{1:n} \leq X_{2:n} \leq \ldots \leq X_{n:n}$ be the sample ordered observations from a distribution with the cumulative distribution function (CDF) $F(x;\mu, \sigma) = F\{(x- \mu)/\sigma\}$ where μ and σ are the location and scale parameters, respectively. For our case we use the Normal distribution as the null hypothesis; however any distribution could be used as the null.

For illustration we will assume that $X_{1:n} \leq X_{2:n} \leq \ldots \leq X_{n:n}$ is from a Normal distribution with an unknown mean μ and unknown variance σ^2. To make the sample linked vectors we need two main components, the length of the ιth vector, and the angle to the horizontal axis θ. Let

$$Y_{i:n} = (X_i - \overline{X})/S \qquad \textbf{2.1}$$

where $\overline{X} = \sum X_{i:n}/n$ and $S = \{\sum (X_{i:n} - \overline{X})^2/(n-1)\}^{1/2}$. $Y_{i:n}$ is now considered the length of the vectors.

Now let $m_{1:n}, m_{2:n}, \ldots, m_{n:n}$ be equivalent to the expected order statistics of the standard Normal distribution, $m_{i:n} = E((X_{i:n}- \mu)/\sigma)$. The angle horizontal to the x-axis is $\theta_i = \pi\Phi(m_{i:n})$ where π = 3.14159… and

$$\Phi(m_{i:n}) = \int_{-\infty}^{m_{i:n}} (2\pi)^{-1/2} e^{(-t^2/2)} dt \,. \qquad \textbf{2.2}$$

Since $m_{1:n} < m_{2:n} < \ldots < m_{n:n}$ and $0 < \Phi(m_{i:n}) < 1$ then $0 < \theta_1 < \theta_2 < \ldots < \theta_n < \pi$. $\theta_{i:n}$ is now considered the angle between the x-axis and the ιth vector.

To obtain the sample linked vectors we use the points $Q_i = (U_i,V_i)$, where $U_i = (\sum_{j=1}^{i} |Y_{j:n}| \text{Cos}\theta_j)/n$ and $V_i = (\sum_{j=1}^{i} |Y_{j:n}| \text{Sin}\theta_j)/n$ are the coordinates of Q_ι (i = 1,…,n) and $Q_0 = (0,0)$. $U_{i:n}$ and $V_{i:n}$ can then be defined as the test statistics, as following

$$U_{i:n} = 1/n \sum_{i=1}^{n} |Y_{i:n}| \text{Cos}\theta_i \,, \qquad \textbf{2.3}$$

$$V_{i:n} = 1/n \sum_{i=1}^{n} |Y_{i:n}| \text{Sin}\theta_i \,. \qquad \textbf{2.4}$$

To obtain the null linked vectors to construct the "known" of the null hypothesis $m'_{i:n} = E(|Y_{i:n}|)$, and $E(m_{i:n})$ are used. To obtain the expected values equation 2.5 must be numerically integrated with the specific n size of the data to generate the values.

$$E(x_{k|n}) = \frac{n!}{(n-k)!(k-1)!}\int_{-\infty}^{\infty} x\left[\frac{1}{2}-\Phi(x)\right]^{k-1}\left[\frac{1}{2}+\Phi(x)\right]^{n-k}\phi(x)dx, \qquad 2.5$$

$$\text{where } \phi(x) = (2\pi)^{-1/2}e^{-x^2/2} \text{ and } \Phi(x) = \int_0^x \phi(x)dx.$$

Using the expected values of $m_{i:n}$ and $Y_{i:n}$ we obtain the point $Q_n = (U_n,V_n)$ which becomes the null-linked vectors endpoint that is the "known", other extensions use this information for analysis.

In a sample that we assume to be Normal, we would expect that the sample linked vectors would closely follow the trajectory of the null-linked vectors, as in Figure 2.1.

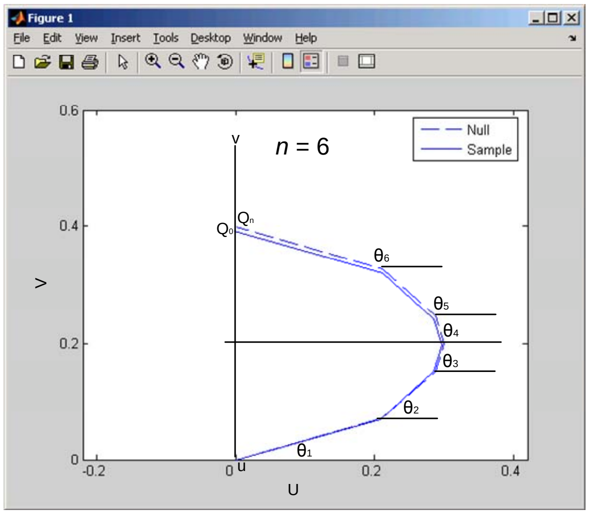


Figure 2.1 – Vector Chart for Normal (n = 6)

However if the sample is not Normal the two paths should differ; how much they differ by depends on the sample distribution. This use of the algorithm allows for an *ad hoc* version (visual inspection) to be used and allow for distribution identification. This *ad hoc* method can be used; however it is rarely done due to the fact that the algorithm generates its own set of test statistics that can with far better accuracy provide an answer to the best fit distribution of a data set in an automated fashion.

Because of the properties of the test statistics generated by the Ozturk algorithm, we can generate a set of equations to model their behavior. Using $m'_{i:n} = E(|Y_{i:n}|)$, we define the expected value for Q_n for the Normal distribution to be $E(Q_n) = E(U_n, V_n) = (0,(1/n)\sum_{i=1}^{n} m'_{i:n}\,\mathrm{Sin}\,\theta_i)$. Symmetric identities for the Normal distribution can be employed to see that $E(U_n) = 0$, and a model can be fitted to V to obtain

$$E(V_n) = 0.326601 + 0.412921/n\,. \qquad \mathbf{2.6}$$

These models can also be applied to other symmetric distributions as well. It is seen that, as long as a distribution is symmetric; $E(U_n) = 0$. With this, models can be provided for the $E(V_n)$ for any other symmetric distribution. Also using more properties of the test statistics and the linked vectors we can obtain models for the variances of U_n and V_n

$$\mathrm{Var}(U_n) = \sigma_u^2 = 0.02123/n + 0.01765/n^2\,, \qquad \mathbf{2.7}$$

$$\mathrm{Var}(V_n) = \sigma_v^2 = 0.04427/n - 0.0951/n^2\,. \qquad \mathbf{2.8}$$

For large values of n it can be shown that the distributions of U_n and V_n are both Normal. Since Q_n is the joint distribution of U_n and V_n, we can apply the bivariate normal distribution to the Q_n endpoint. Using the exponent component of the general bivariate normal we have

$$\frac{1}{2(1-\rho^2)}\left[\frac{(U_n-\mu_u)^2}{\sigma_u^2} - 2\rho\frac{(U_n-\mu_u)(V_n-\mu_v)}{\sigma_u\sigma_v} + \frac{(V_n-\mu_v)^2}{\sigma_v^2}\right], \qquad \mathbf{2.9}$$

where μ_1 and μ_2 are the means, σ_1^2 and σ_2^2 are the variance of U and V respectively, and ρ is the coefficient of correlation between U_n and V_n. Under the assumption that μ_u is always 0 and that value of ρ between U_n and V_n is 0, we obtain the simplified version of equation 2.9,

$$\frac{U_n^2}{\sigma_u^2} + \frac{(V_n-\mu_v)^2}{\sigma_v^2} = -2\ln\alpha\,, \qquad \mathbf{2.10}$$

where α is the confidence level. For any given sample size, μ_v, σ_u^2, and σ_v^2 can be obtained from equations 2.6, 2.7 and 2.8, respectively. With these values equation 2.10 can be solved for α, and return the specific α value for each set of endpoints of a given data set. Setting equation 2.10 to a set of ellipses, generated by varying the value of α, and putting it onto the null endpoint Q_n, we can create a set of 100(1- α) % confidence ellipses around the Normal distribution. With these confidence ellipses we can now set thresholds on the sample data in order to differentiate between potential Gaussian data, or some other distribution. In Figure 2.2, we can easily see that all three data sets plot within the 99% confidence ellipse. However the single data set falls outside of the 95% confidence ellipse, indicating that there is more than a 95% chance that, the distribution is not normal, and most likely another distribution. The other two data sets plot relatively close to the null endpoint of the Normal distribution, so the assumption is that the best fit distribution for these is Normal.

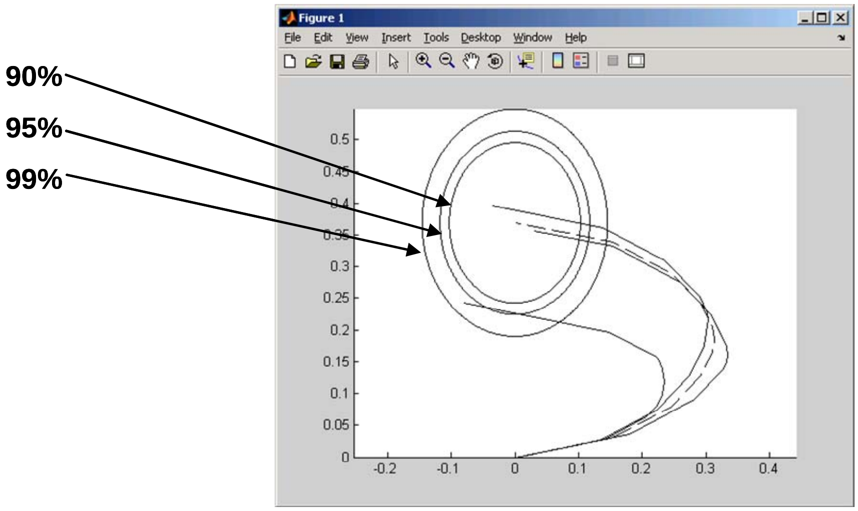


Figure 2.2 – Sample Data with Confidence Ellipses

As before with the fitted models for the $E(V_n)$, and the variances of U_n and V_n, other distributions can have confidence ellipses fitted to themselves. Currently the only distributions that can have confidence ellipses fitted to them, are those that have U_n and V_n values that model a bivariate Normal distribution. When this can be proven, the values required to be used in equation 2.9 can be calculated and provide the α values for that distribution, and set up the confidence ellipses as well.

General reference is made to [4]. Another of the powerful tools provided by the algorithm is the ability to estimate the location and scale parameters or the approximated PDF. The two main test statistics come from linear combinations of equations 2.3 and 2.4,

$$T_1 = a\alpha + b\beta\text{,} \tag{2.11}$$

$$T_2 = c\alpha + d\beta\text{,} \tag{2.12}$$

where $a = \sum_{i=1}^{n} Cos(\theta_i)$, $b = \sum_{i=1}^{n} \mu_{i:n}\mathrm{Cos}(\theta_i)$, $c = \sum_{i=1}^{n} Sin(\theta_i)$ and $d = \sum_{i=1}^{n} \mu_{i:n}\mathrm{S}in(\theta_i)$. However for the Normal distribution it can be shown that a = 0. So for any time that the null distribution is chosen to be Normal, a will always be zero. For a = 0 we obtain

$$\widehat{\alpha} = (T_2 - d\widehat{\beta})/c\,, \qquad \mathbf{2.13}$$

and

$$\widehat{\beta} = T_1 / b\,, \qquad \mathbf{2.14}$$

which are unbiased estimators of α and β respectively. For any symmetric distribution equations 2.13 and 2.14 can be simplified further, by the fact that d = 0 for symmetric distributions under the null hypothesis of Normal. Therefore we obtain

$$\widehat{\alpha} = T_2 / c \qquad \mathbf{2.15}$$

and

$$\widehat{\beta} = T_1 / b\,. \qquad \mathbf{2.16}$$

Though these are simplified further we use, equations 2.13 and 2.14 to provide for a generalization of all distributions under the Normal null hypothesis.

The shape parameter of a distribution can also be calculated for the best fit distribution. The algorithm has the potential to calculate parameters for distributions that have one and two shape parameters. However this advanced function was left out due to the fact that no research into shape parameter distributions has been conducted in this effort. The procedure for explaining how to derive the estimators is contained in [4].

2.2 Extension of Features – Ozturk Algorithm

Because of the versatility of the Ozturk algorithm, it lends itself the ability to modify its exact output. Previously the algorithm was for the most part *ad hoc* and had no need to alter its output. However by displaying the information the Ozturk algorithm returns to us, we can create multiple tests. The first of such modifications was created last summer in the form of the new plotting techniques. This summer, two more extensions are brought out by the algorithm using the same statistics that create the plots.

2.2.1 Extension – Plotting Techniques

For the most part graphing utilities of the algorithm were lacking. One of the most accurate identification charts, shown in figure 2.3, shows at least visually what the chart should look like. However the true accuracy of this older chart is in question due to the fact that most of the values do not line up correctly. Though this may be true, we can at least use it as a template for what future charts should trend to.

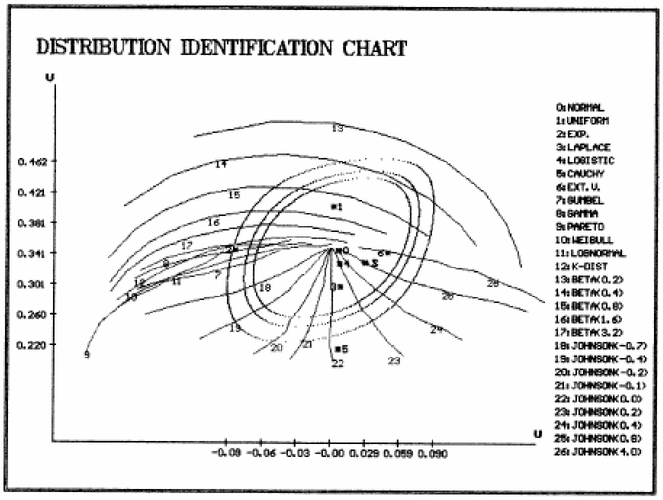
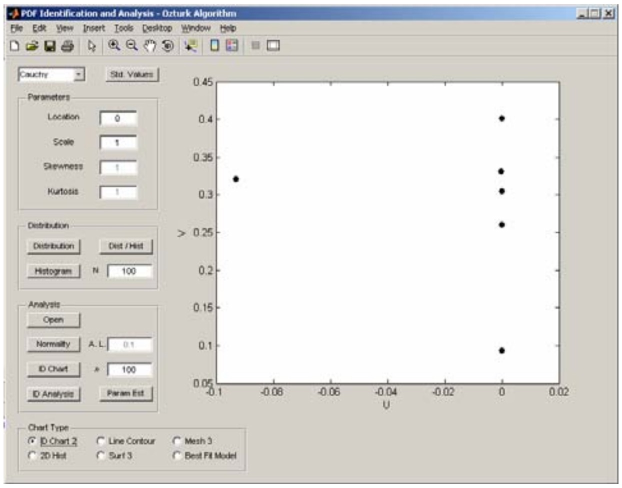


Figure 2.3 – Distribution Identification Chart (Old)[1]

The first effort was to try and replicate the results provided in the identification chart. Since no distributions with shape parameters have been implemented into the code in this effort, only the integrity of the location-scale family distributions could be tested. The only reliable method of testing this was to make certain that the points for the distribution match the expected values of the fitted models for the distribution. This was easily done by simple Monte Carlo simulations, and finding the mean endpoint of 100,000 runs. For each of the six distributions that we have, we were able to recreate the plot, show in figure 2.4, and allow for a very high accuracy as compared to the fitted values. Now that we were able to recreate the basic identification chart of the algorithm; we could now advance the concept. In order to do this we look back towards the idea of *U* and *V* both being from some distribution, and *Q* being the joint distribution of *U* and *V*. So *Q* must then be a bivariate distribution, in which case the current distribution identification charts do not have the power to correctly display *Q* as that bivariate distribution. In order to better portray the relationship of *U* and *V*, we move the graphs from two-dimensional to three-dimensional. And since each distribution has a set of U and V values, each distribution has a bivariate distribution to accompany it. We are then able to recreate the distribution identification chart on a three-dimensional graph now, as shown in figure 2.5.



Cauchy

Location 0
Scale 1
Skewness 1
Kurtosis 1

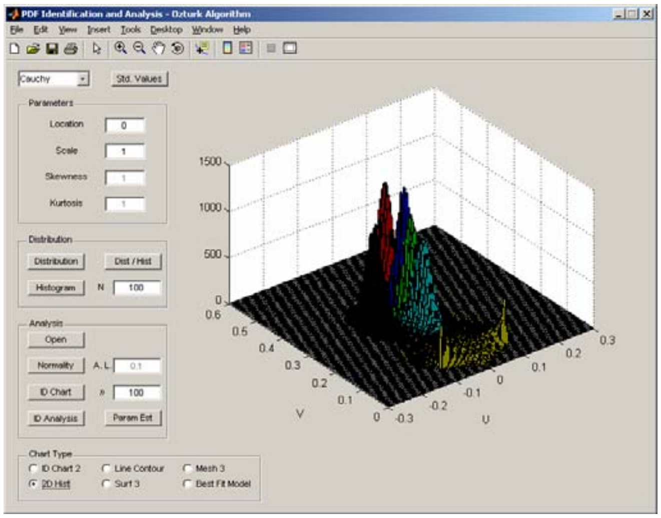
N 100

A.L. 0.1

n 100

● ID Chart 2

Figure 2.4 – Distribution Identification Chart (New)



Cauchy

Location 0
Scale 1
Skewness 1
Kurtosis 1

N 100

A.L. 0.1

n 100

● 2D Hist

Figure 2.5 – 3-D Distribution Identification Chart

The bivariate distributions are now much easier to see, because in the most basic sense the plot is simply a histogram designed for bivariate distributions instead of univariate distributions. Also since all the data is stored into a matrix form, it is easy to manipulate the data into different forms, as seen in figure 2.6. Each of the plots in figure 2.6 displays the same information, which is represented in figure 2.5, excluding the bottom right graph. However the data is plotted in different manners to allow for personal preferences to select which plot more clearly displays the information for them. The bottom right graph has only four distributions on it for a reason. The graph is called the best fit model, which displays the Uniform, Normal, Laplace and Logistic bivariate distributions. Since each of those four seems to have bivariate Normal distributions, this is what each of the distributions is modeled after. The Exponential and Cauchy distributions do not appear on the best fit model graph because of the fact that their bivariate distribution is not Normal.

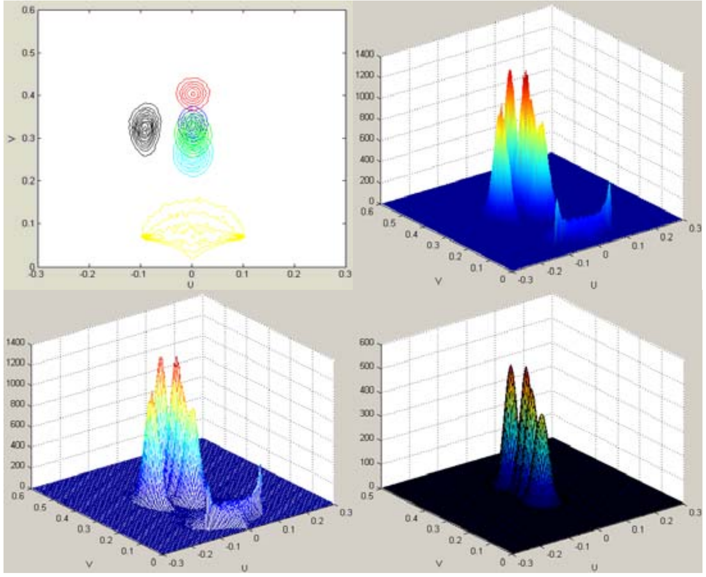
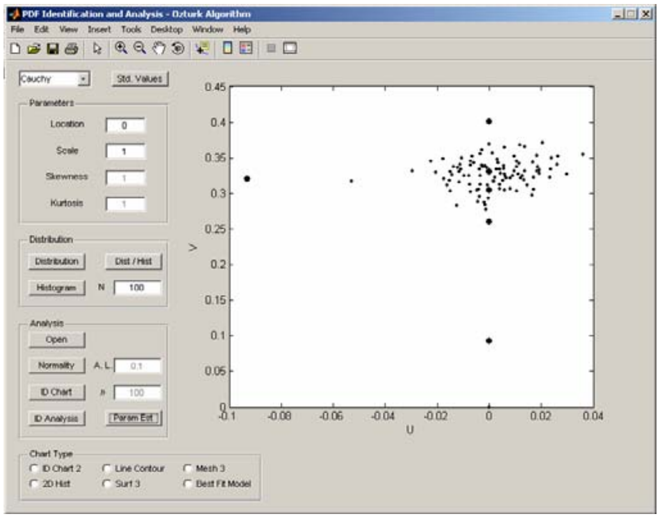


Figure 2.6 – Multiple 3-D Distribution Identification Charts

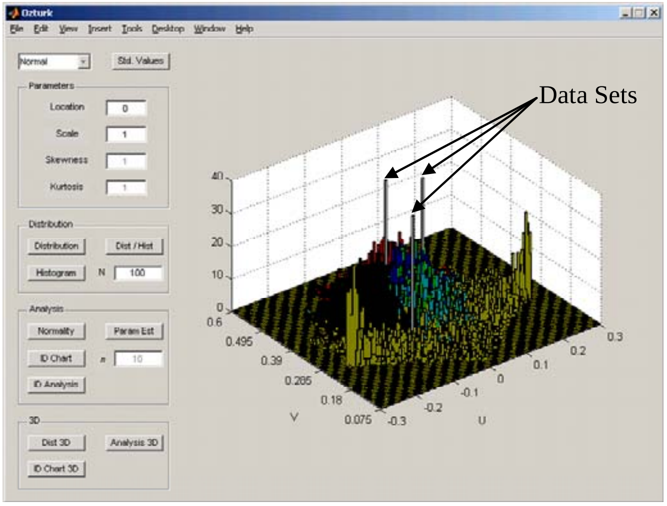
Since part of the algorithm is *ad hoc* it makes it much easier for the user to see the best fit distribution when the data can be plotted onto a specific identification chart. To best do this, each identification chart will have to display the data in a different manner to be able to observe it. The first step is to apply this to the regular identification chart. This is simply done by plotting the endpoints of each of the data sets onto the chart, as seen in figure 2.7. At the minimum the analysis of this chart gives you a decent estimate as to which distribution would best fit the data. However when identifying the distribution this chart is based on distance away from a null endpoint instead of any sort of test. The next two types of charts are based on the new plots, specifically the bivariate histogram chart in figure 2.5 and the line contour chart in the upper left corner of figure 2.6. Using the bivariate histogram chart, the easiest method to implement the data was to make “flag poles” at the points on the chart at which the data plotted to, shown in figure 2.8.



Cauchy

Location 0
Scale 1
Skewness 1
Kurtosis 1
N 100
A. L. 0.1
n 100

Figure 2.7 – Regular ID Chart with Data



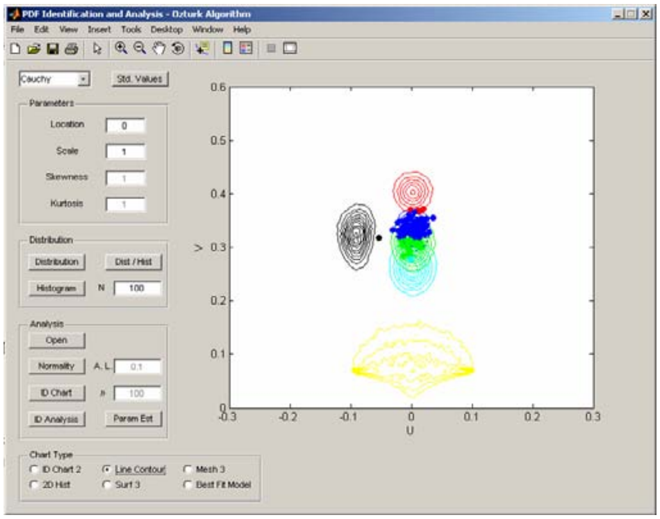
Normal

Location 0
Scale 1
Skewness 1
Kurtosis 1
N 100
n 10

Figure 2.8 – Bivariate Histogram with Data[6]

This chart, still purely *ad hoc*, allows the user to better fit a distribution to the data. However there is a drawback to this chart since it loses power on lower data size values. The last chart is no longer purely *ad hoc* as it uses a likelihood ratio test to determine the best fit distribution and then plots that point the same color as the distribution appears on the identification chart, shown in figure 2.9. Of the three charts the third does the best visually and has the added power of the likelihood ratio test to automatically classify the data for the user.

Each of the new plots extends the ability of the algorithm to visually identify distributions. To increase the power of the analysis charts a likelihood ratio test could be applied to the first two charts or the alpha levels of each point could identify the distribution. However each of these charts was implemented prior to this summer, so the main focus was not directed towards them, but towards the two new extensions.



Cauchy

Parameters

Location 0

Scale 1

Skewness 1

Kurtosis 1

N 100

A. L. 0.1

n 100

Chart Type

○ ID Chart 2 ● Line Contour ○ Mesh 3

○ 2D Hist ○ Surf 3 ○ Best Fit Model

Figure 2.9 – Line Contour Chart with Data

2.2.2 Extension – Correlation Test

Since the Ozturk algorithm only handles uncorrelated data, what exactly occurs when the data is correlated? Because it is sensitive to correlation, we have been able to use this sensitivity to detect correlation within the data that is being analyzed. The test that has been developed is specifically a test to determine whether or not a set of Gaussian data is correlated or uncorrelated, and if it is correlated, to what degree. This allows us to decide whether or not the results from the algorithm are acceptable or are too skewed by correlation.

Before this test could have been proposed, the main algorithm had to be tested to observe how correlation affected it. In order to carry out a controlled experiment, a set of filters were chosen within Matlab to create correlated samples. The filters chosen were bandpass and lowpass finite filters, Butterworth bandpass and lowpass filters and Chebyshev Type 1 and Type 2 lowpass filters. For each specific filter a set of filters were designed, with a certain bandwidth and filter length, and in the cases of the infinite filters, other certain parameters. For each filter, 100,000 sets of 100 data values from a Normal distribution were sent into the filters and then run into Ozturk. From there the statistics are plotted and the observations are saved so that a compilation of each filter could be created. A set (the same size as the Normal set) of Exponential and Uniform were also run and compiled.

Looking at each of the plots, one could tell that correlation, evenly moderately weak correlation seemed to throw off the identification ability of the algorithm. Also there was no noticeable pattern for each specific distribution, in that after a certain amount of correlation the three distributions began to appear very similar. However, one pattern was noticed within all the distributions for the lowpass finite filters. As the bandwidth decreased and the filter length increased, a definite shape was being defined, seen in figure 2.10. Each of the filters though produces different results and no comparison between them could be made.

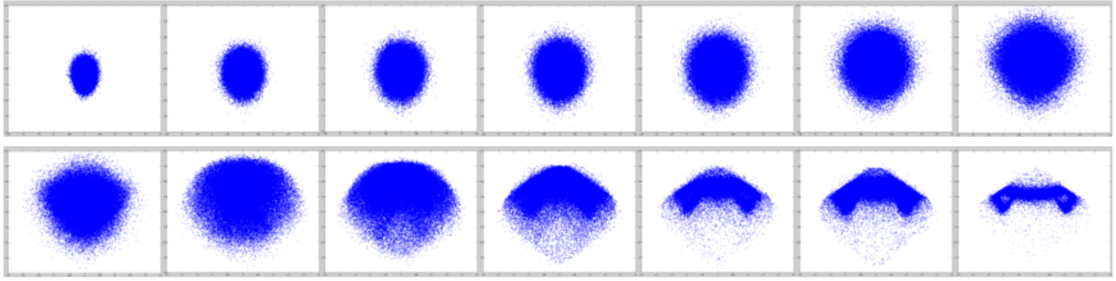


Figure 2.10 – Finite Lowpass with Normal Data

Now with these observations we could go ahead and try to create a method to read the data into the algorithm and have it realize that the data is correlated or uncorrelated. The first approachable method to doing this was to simply decimate the data. By constructing a segment of Normal filtered data that was 100,000 points in length we could decimate the data to every 100th point. This data is less correlated simply because the decimation will be further in length then the finite filter length. However when the filter length increases the data has the tendency to revert back to being correlated once again. The decimation of the data sample works for small filter lengths, but does not possess the abilities or features that we desire in order to test for correlation within the algorithm.

The next test was to take the FFT of the sample then process the real and imaginary parts as two separate data sets. So if we started with a set of data that had 100,000 runs with a size of 100, we would return 2 sets of data that had 100,000 runs of 49 (we exclude the first value which is the mean). From the observations of just a single data set with correlated and uncorrelated data, we can see a distinct difference in the FFT of the data, shown in figure 2.11. So again we take all of the filters and parameters and reconstruct all the data sets once more, take the FFT of the data sets and plot the two graphs, the real and the imaginary.

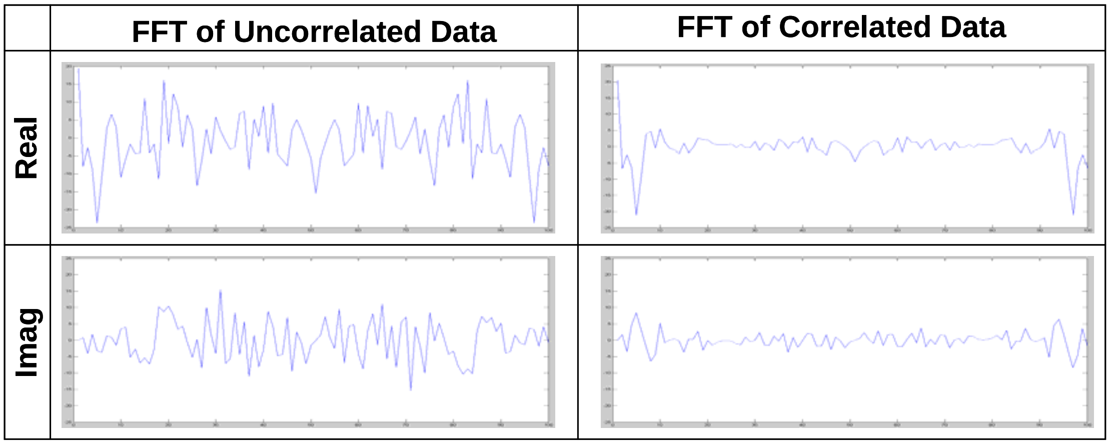


Figure 2.11 – FFT of Uncorrelated/Correlated Data

As we go through all the observations that were produced by the test data, patterns have become increasingly clear among the Normal distribution specifically. The major point being that it does not matter which filter is being used, each takes on the same form, given similar parameters. For both low pass and high pass filters there is a specific pattern and among all bandpass filters there is also a pattern. As far as the null, or no correlation present observation, the FFT of a Normal distribution does not change its distribution so in the end it again plots Normal. So when correlation is introduced the real and imaginary plots trend away from the Normal and to specific shapes. Looking at the lowpass observations, shown in figure 2.12, we see, as the bandwidth decreases and the filter length increases, patterns appear. We can also set these observations to a high pass filter with the same parameters, and the results appear the same. In the bandpass case we took a sliding window across the normalized bandwidth of 0.05 radians, all with the same filter length. (The numbers in brackets represent the normalized frequencies as Matlab filter parameters.) The results, shown in figure 2.13, show us that there is symmetry around the center bandwidth. What helps the observations further is that when the bandpass filter approaches a low pass or high pass filter, it begins to take the form of a low pass or high pass filter in the plots.

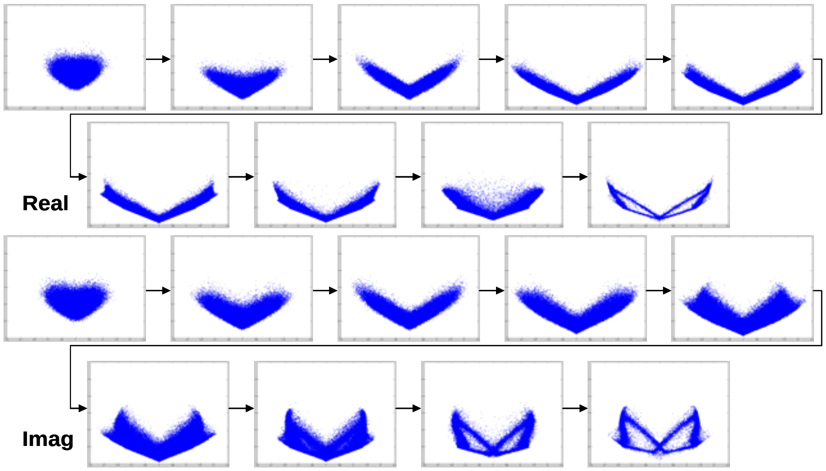


Figure 2.12 – Low pass FFT Real and Imaginary Plots

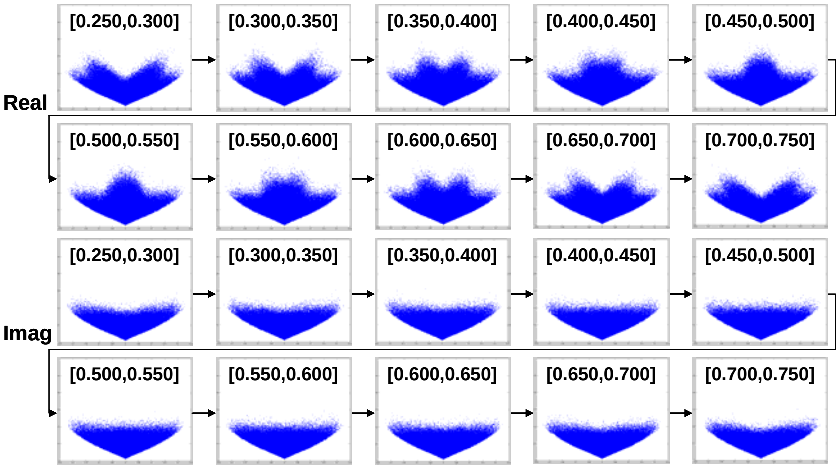


Figure 2.13 – Bandpass FFT Real and Imaginary Plots

Observe that no correlation is the same as the null Normal point, and when correlation is added the point moves in a set path away from the Normal point. This allows us to simply set the correlation test as a test for Normality. If the data all falls outside of the confidence ellipse for Normal, and we were testing Normal data to begin with, then the original data must be correlated. By doing this we can perform various tasks. First since the amount of correlation generally plots the real and imaginary data to a certain area on the graph with a certain shape, we can estimate the amount of correlation present in a given sample. Vice versa, if we are given the amount of correlation we can estimate where the data will plot onto the two plots.

Because we can vary the confidence ellipse by simply changing the value of α we can generate a set of ROC curves in order to show the performance of the new test. To generate the curve we simply take a set of uncorrelated data and calculate how many points out of 100,000 falls within a certain α level. This gives us the values of the x-axis which is the false alarm probability. To get the y-axis values we take different sets of correlated data and run the same test. The ROC curve is shown in figure 2.14, with three different sets of correlated data. Each of the three sets are weakly correlated sets, however the test picks up the correlation within the samples. (The numbers in parenthesis represent Matlab filter parameters.)

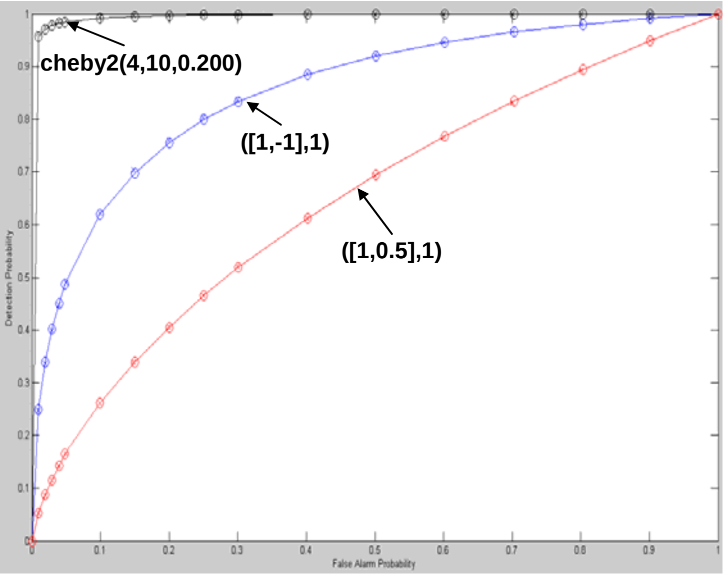
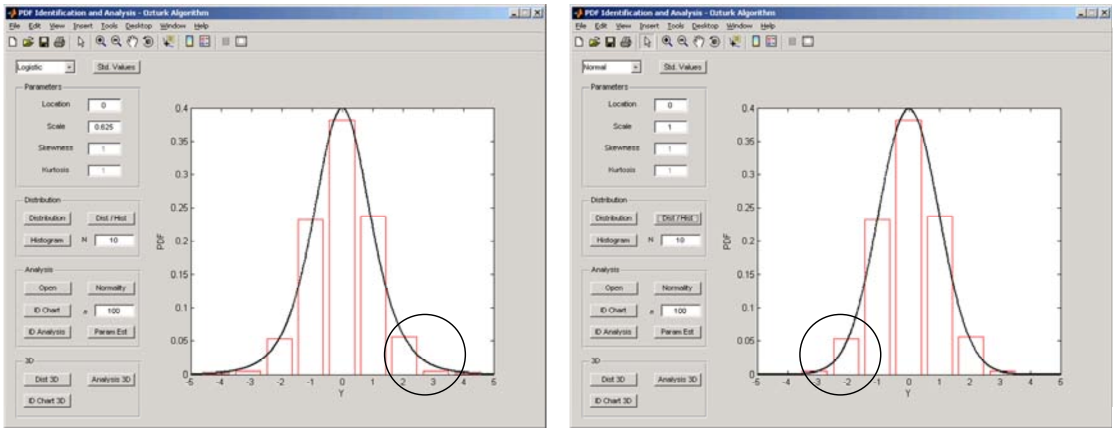


Figure 2.14 – ROC Curve of Correlation Test

The reasoning behind the test only being able to detect correlation in Gaussian data is mainly due to the method we use to find it. Once we take the FFT of most distributions they appear Normal instead of whichever distribution they were originally. This poses the problem. If on the main plot the data looks Normal but isn't, and then plots to Normal on the real and imaginary plots, the assumption is it is uncorrelated Normal data. Also since there was a large amount of research effort this summer taken to test the Normal distribution under this new test, there was not enough time to take the same amount of observations from other distributions. However from the few observations that were made, we can conclude that for weak correlation the distributions follow the Normal on the FFT plots. However, when stronger correlation is introduced, the other distributions trend away from the Normal.

2.2.3 Extension – Multi-Distribution Identification

One of the problems with most distribution identification tests, if not all, is what happens when you have a distribution mimicking another? When you identify the best fit distribution is it the original distribution or is it the mimicked distribution? As shown in figure 2.15, there are numerous distributions that can be modified by their parameters in order to form another distribution. In most cases those distributions with a shape parameter tend to always have at least one value for the parameter to turn the distribution into another.


Logistic
Location 0
Scale 0.625
Skewness 1
Kurtosis 1
N 10
n 100

Normal
Location 0
Scale 1
Skewness 1
Kurtosis 1
N 10
n 100

Figure 2.15 – Laplace(0,0.625) v Normal(0,1)

To help alleviate this problem we have set up a type of multi-distribution identification where the best fit distributions are found not just a single distribution. To do this we simply find the top three best fit distributions to a data set. From there we can calculate the parameters for each of the best fit distributions. This now allows the user to select which distribution they believe is the best, along with the help of the algorithm. So when there is a set of distributions that are all relatively close in test values, then each of these distributions could fit the data. However if it ends up that one distribution has a much higher value then the others, it is most likely the first.

As of right now the new feature works and does return the top three distributions based on the likelihood ratio test. This allows the user themselves to choose the best fit distribution for their data upon inspection of the information provided by the algorithm. However there are some implementation issues in the new feature that need to be taken care of, but they deal mostly with the ability of the algorithm to estimate the parameters of the data.

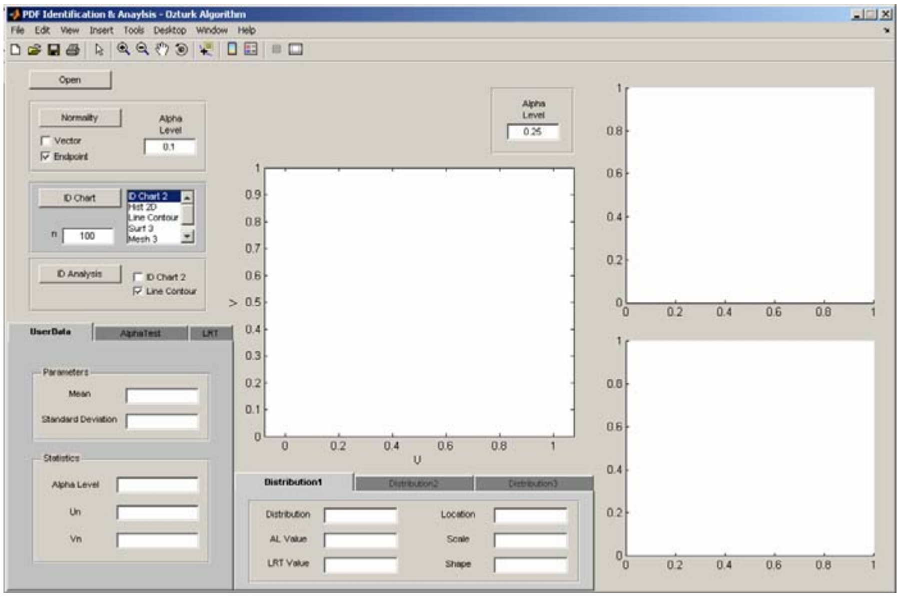
3.0 GUI Implementation

With two entirely new features to the algorithm 3-D distribution display and correlation display, the old GUI would not be able to support the new tests. In order to house these new features a new GUI was made. The new GUI left out some features of the previous interface, but expands its capabilities. Again the GUI was created in Matlab and used various tools crafted by the community file exchange of Mathworks. Most of the old code could be salvaged and was just updated to run with the new interface.

In the main GUI window, as seen in figure 3.1, there are now three plots to allow for both FFT plots and the original plot, along with two sets of tabs to house both the user data and the best fit distributions. The GUI functions have been split into five sections, the Normality test, the identification charts, the analysis with the identification charts, the correlation test, and user information tabs. Once the Open pushbutton is pressed the user selects the file which they will be analyzing, and the code proceeds to process all the information that it will need to display for a set of data.

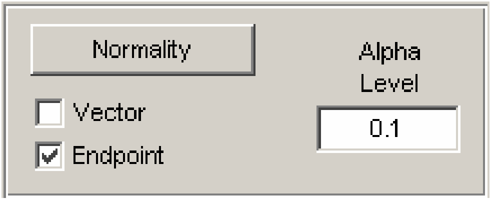
3.1 GUI – Normality Test

The Normality test panel consists of various options in order to display the data in the best manner for the user. Shown in figure 3.2, we see that the panel consists of aNormality pushbutton, Vector and Endpoint checkboxes and an Alpha value text box. By choosing the Normality button, the code plots the sample data, with the null linked vector of the Normal and the confidence ellipses, seen in figure 3.3. The Normality pushbutton also proceeds on with the correlation test. The function and area of the GUI that deals with the correlation test will be discussed in section 3.4. The data is set to plot as endpoints instead of vectors when the GUI opens. However to change the plot so that the vectors appear, the user only has to select that checkbox. Once selected the user can return to the endpoint plot by selecting the Endpoint checkbox. The Alpha level value box can be changed by the user to be a value in between zero and one. Once the user has chosen a value the innermost ellipse will plot to that specified value. Also the data that falls within that ellipse is plotted in red, so the data will be updated as the Alpha value changes as well.



☐ Vector
☑ Endpoint
Alpha Level 0.1
n 100
☐ ID Chart 2
☑ Line Contour
Alpha Level 0.25

Figure 3.1 – Updated GUI Main Window



☐ Vector

☑ Endpoint

Alpha Level 0.1

Figure 3.2 – GUI Normality Test Panel

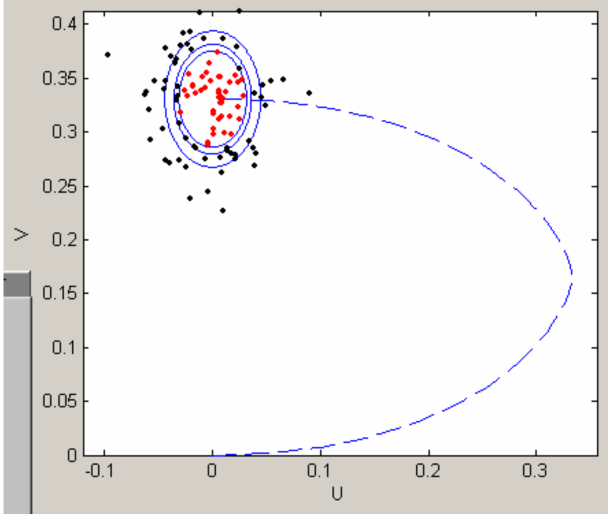
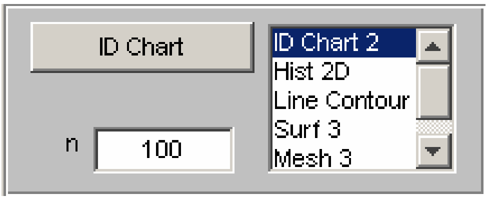


Figure 3.3 – GUI Normality Test Plot

3.2 GUI – Identification Charts

The identification charts panel contains the regular identification chart and the new charts produced last summer. Within the panel are the chart choices, the specified n size, and the ID Chart pushbutton, shown in figure 3.4. At any point during the use of the GUI the user can select a specific n size and plot the identification chart that they would like to see. If the data for the chart does not exist already in memory the algorithm generates the values for the chart. However the computational time of generating these graphs is still relatively high due to the amount of data it has to train on.

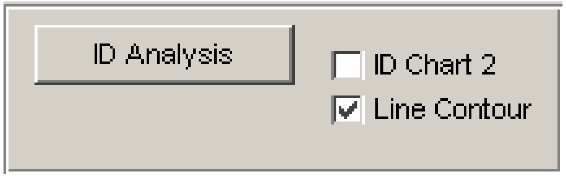


n 100

Figure 3.4 – GUI Identification Charts Panel

3.3 GUI – Analysis with Identification Charts

The analysis panel contains two of the identification charts with data plots that were selected due to their simplicity and ability to display the most information. Within the panel, as in figure 3.5, we see the two choices: the ID Chart 2 and the Line Contour charts.



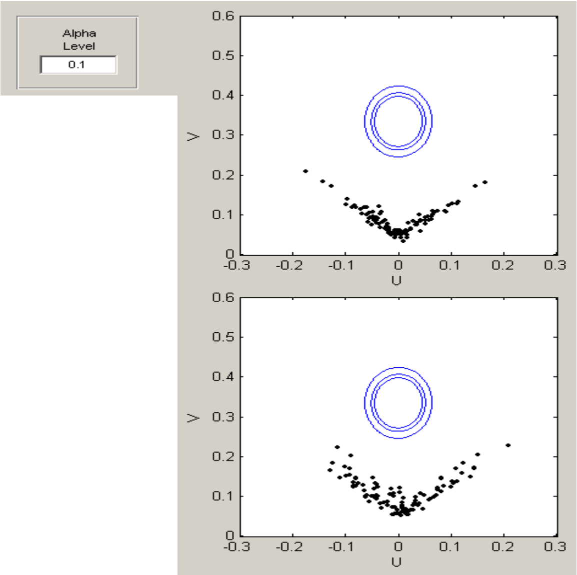
□ ID Chart 2
☑ Line Contour

Figure 3.5 – GUI Analysis Panel

Just as before the ID Chart 2 takes both the regular identification chart and the endpoints of each data set and plots the results on the same set of axes. Also the Line Contour chart takes the line contour identification chart and the endpoints of the data, and plots each of the endpoints the same color as the distribution it has been identified as.

3.4 GUI – Correlation Test

The right hand side of the GUI deals with the correlation test. Once the Normality test is chosen, the correlation test proceeds by plotting the data onto the two side graphs for analysis. In its most simplistic nature the correlation test is simply a test for Normality of the real and imaginary sets of FFT data. The idea behind this is the fact that when no correlation is present the data should be Normal, and when correlation is presented, the data deviates from the Normal region. In the GUI itself the real FFT data plots on the upper graph and the imaginary plots on the lower graph, shown in figure 3.6.



Alpha Level 0.1

Figure 3.6 – GUI Correlation Test Plot Area

Also for the two correlation test plots, there is a edit value alpha level. This is there such that the user can change the alpha level to change the detection and false alarm rate of the test. The test is presented within the Normality test to show that when the data is classified as being correlated, it cannot be used to correctly identify the best fit distributions of the data. However if it is uncorrelated, then the data can be continued being analyzed.

3.5 GUI – User Information Tabs

In order to display back the most information that the algorithm uses, two sets of tab panels were implemented into the GUI, shown in figure 3.7. On the left most tab panel is where the user data and the test values go. The user data tab displays the sample mean, sample standard deviation, alpha level with respect to the Normal distribution, and the *U* and *V* endpoint values of that specific data set. The second tab, labeled AlphaTest, returns to the user the results of the Alpha Test. In this test the algorithm returns the values of the alpha values with respect to the Normal, Uniform, Laplace and Logistic distribution. The third panel, labeled LRT, returns the user the results of the likelihood ratio test. This second classification test simply takes the values of the matrices that are used to plot the new identification charts. Once the endpoint is found, the algorithm goes to that endpoint in the matrices, and returns that value for each specific distribution. The second set of tabs displays the top three distributions that the algorithm identifies. Such that the left most tab is the best fit, the middle tab is the second best fit and the right tab is the third best fit distribution. Within each of the tab panels the GUI returns to the user the distribution name, the Alpha Test result for that specific distribution, the likelihood ratio test for that specific distribution and the estimated location, scale and shape parameters. The identification of the best fit distributions is based on the likelihood ratio test, so for a check as you move left to right on the tabs, the LRT values should decrease in value.

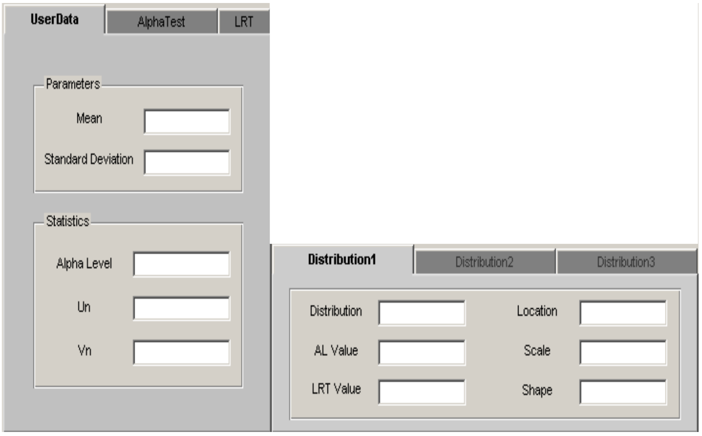


Figure 3.7 – GUI User Information Tabs

In order to get any of this data for a specific endpoint the user must only click on that data point in one of the graphs that the data is plotted. These specific graphs include both of the Normality plots, linked-vectors or endpoints, and both of the analysis charts, regular identification chart and line contour chart. Any point or vector within the main chart area for each of these plots can be simply clicked on to obtain the user information for that specific endpoint.

4.0 Future Work

In order to make the algorithm more efficient at identifying distributions the next step would be to increase its distribution library. Currently the implementation only contains six location – scale distributions which are, Normal, Uniform, Laplace, Logistic, Exponential and Cauchy. It would benefit the implementation to have a much more expansive library of distributions and include distributions that have shape parameters associated with them also. Though adding more location – scale distributions may be an easy task, by just simply creating data through Monte Carlo simulations. However the shape parameter distributions will offer much more difficulty. For each value of a shape parameter the distribution will have a set of values, and this shape parameter much be varied in order to get all possibilities of the distribution. This forms a line on the identification chart. Although possible, the amount of data that will be needed to run for a single specified *n* value will be more then what is run in the entire algorithm as of now, and could be too time consuming.

Other work would be to include furthering the abilities of the correlation test. One of the most applicable extensions to the correlation test that can be seen as of right now, is finding a closed form bivariate distribution that models the solution to our correlation observations if one exists. In both figure 2.12 and 2.13 we have been able to notice a predictable pattern among the correlation for all filters that were tested. The basis for our observations that there might be a bivariate distribution to describe these events is based on that predictable pattern. When there is no correlation present the distribution will start at the bivariate Normal distribution. As correlation is steadily added to the data the distribution moves in a certain pattern. So the theory that has been adopted so far is that this distribution has a bivariate Normal equation within it, that when certain parameters are zero, returns to you the bivariate Normal. Also if this solution was found it may lead to some type of parameters that can be measured. If these parameters can be measured, we then have a statistic that can be calculated for the amount of correlation present.

Work that can also been done with the correlation test includes the analysis of other starting distributions. In such a case we would start with some other distribution such as Uniform and run the same set of tests that we did for the Normal data. If we are able to notice patterns in these distributions as well, there is the possibility of multiple tests. Either each distribution will be able to be identified separately or the test will be generalized to testing for correlation within any type of sample. To improve upon this technique we may also seek alternatives to the FFT technique that we are currently using. As a replacement we may insert either a SVD or Matrix Pencil technique to try and increase test capabilities.

Other potential work will be to look at the q-Gaussian distribution. The q-Gaussian distribution is described simply as a Normal distribution with a shape parameter. Also more work on the multi-distribution identification idea could be conducted in order to improve accuracy. An increase capability in calculating the parameters of a distribution with better accuracy would grant more power to the identification technique.

4.1 Acknowledgements

The author would like to once again extend his gratitude to Dr. Andrew Noga for his continued mentoring. Dr. Noga once again created the general task for the summer and allowed for enough freedom to allow for this work to progress as successfully as it did. A thanks is also extended to those that provided additional comments on the algorithm and data that was provided to help proof the resourcefulness of the algorithm.

4.2 References

[1] Ozturk, A. An application of a distribution identification algorithm to signal detection problems. Signals, Systems and Computers, 1993. 1993 Conference Record of the Twenty-Seventh Asilomar Conference on 1-3 Nov. 1993 pp.248-252 vol. 1.

[2] Ozturk, A. and Dudewicz, E.J. A new statistical goodness-of-fit based on graphical representation. Biometrical Journal vol. 34 ($), pp.403-427, 1992.

[3] Ozturk, A. A general algorithm for univariate and multivariate goodness-of-fit tests based on graphical representation. Communications in statistics-theory and Methods. Vol.20 (10), pp.3111-3137, 1991.

[4] Ozturk, A. A new method for univariate and multivariate distribution identification. Pre-publication, pp.1-14, >1990.

[5] Salerno, S. A Matlab-Based Ozturk Algorithm Implementation. DTIC Report #ADA450188, May 2006.